UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/501,845	08/26/2005	Martin Vorbach	2885/86	9148

26646 7590 04/11/2011

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
04/11/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/501,845	VORBACH ET AL.
	Examiner	Art Unit
	KEITH VICARY	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 March 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *7-13,15-17 and 19-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *7-13,15-17 and 19-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)	
1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *10/28/2010, 1/7/2011, 2/3/2011*.	6) ☐ Other: _____.

DETAILED ACTION

1. Claims 7-13, 15-17, and 19-24 are pending in this examination and presented for examination. Claims 7 and 23-24 are newly amended by amendment filed 3/28/2011.

Claim Objections

2. Claim 23 is objected to because of the following informalities. Appropriate correction is required.

3. In claim 23, lines 14-15, "the reconfigurable field" should be "the reconfigurable field of data processing cells" for antecedent basis purposes.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 7-13, 15-17, and 19-24 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

6. Claim 7 recites the limitation "wherein at least one of: (a) a single function of the program is executed via sequential execution of a plurality of the set of configurations;

and (b) for each of at least one of the set of configurations, a plurality of functions of the program are executed in sequence via execution of a single instance of the respective configuration" in lines 29-34. However, the original disclosure does not appear to disclose this limitation (e.g. page 5, lines 30-32, discloses that the traditional instruction [and not a plurality of instructions] is replaced by a configuration [and not a plurality of configurations] in the known sense, referred to in the following discussion as a complex instruction word).

In view of the possibility that applicant believes that the aforementioned newly amended limitations have support due to the incorporating by reference of PACT11 (as was argued in regards to claims 23 and 24, which contains similar limitations and were previously present), examiner notes that the response to arguments regarding the written description rejection of claims 23 and 24, as detailed below, are relevant to the aforementioned limitation of claim 7.

7. Claim 7 recites the limitation "transmitting data directly from any one of the data processing cells to any other of the data processing cells" in lines 7-8. However, the original disclosure does not appear to support this concept of transmitting data "directly" from "any one" of the data processing cells to "any other" of the data processing cells.

a. Claims 8-13, 15-17, and 19-22 are rejected for failing to alleviate the rejection of claim 7 above.

8. Claim 23 recites the limitation "an instruction of the program is executed using a plurality of the configurations" in line 14. However, the original disclosure does not

appear to disclose this limitation (e.g. page 5, lines 30-32, discloses that the traditional instruction is replaced by a configuration [and not a plurality of configurations] in the known sense, referred to in the following discussion as a complex instruction word).

In response to examiner's previous recommendation that applicant specifically point out where and how in the original disclosure this limitation is supported in the event that applicant disagrees with the rejection, applicant argues that:

"The present application, at par. 37 of the application publication no. 2006/0090062, incorporates by reference internal docket number PACT11, which includes U.S. Pat. App. Ser. No. 09/967,847 (issued as U.S. Pat. No. 7,210,129), which claims priority to German Pat. App. No. DE 101 39 170, which is itself separately referred to and incorporated by reference at par. 3 of the application publication. (See also U.S. Pat. App. Pub. No. 2009/0006895, which explicitly states that PACT11 includes U.S. Pat. App. Ser. No. 09/967,847.) Support for the above-noted features of claims 23 and 24, respectively, may be found, for example, in U.S. Pat. No. 7,210,129 at 3:16-19, 3:33-35, 4:23-25, 6:41-42, 7:51- 58, and 11:1-6."

However, there are numerous deficiencies in this argument. First, the original disclosure does not appear to provide support for applicant's argument that PACT 11 "includes U.S. Pat. App. Ser. No. 09/967,847 (issued as U.S. Pat. No. 7,210,129)" because the IDS dated 7/16/2004 merely correlates "PACT11" to "DE 101 39 170.6". Applicant cites U.S. Pat. App. Pub. No. 2009/0006895 as explicitly stating that PACT11 includes U.S. Pat. App. Ser. No. 09/967,847. However, this explicit statement is not part of the original disclosure. Examiner further generally notes the confusion that

arises in assigning same "internal reference numbers" to different references in different pending patent applications.

Second, examiner notes that "DE 101 39 170.6" does not appear to provide support for the limitation in question. Examiner's conclusion is based on a translation of the entirety of "DE 101 39 170.6", included with the current office action. Given that more than 30 DE references are incorporated by reference, examiner respectfully asks applicant to provide translations and specific citations for any DE references relied upon to support subject matter in order to further compact prosecution. Examiner further generally notes that in any application which is to issue as a U.S. patent, essential material (defined as that which is necessary to (1) describe the claimed invention, (2) provide an enabling disclosure of the claimed invention, or (3) describe the best mode) may not be incorporated by reference to (1) patents or applications published by foreign countries or a regional patent office, (2) non-patent publications, (3) a U.S. patent or application which itself incorporates "essential material" by reference, or (4) a foreign application.

Finally, examiner notes that U.S. Pat. No. 7,210,129 does not appear to provide support for the limitations (even though U.S. Pat. No. 7,210,129 is not properly incorporated by reference, as detailed above). For example, while U.S. Pat. No. 7,210,129 may disclose "execution of a WHILE loop according to Fig. 4b over an number of configurations" in col. 11, lines 1-6, this disclosure is in the context of the invention of U.S. Pat. No. 7,210,129, and there is no linking support that an instruction of the program [of the instant application] is executed using a plurality of the

configurations [of the instant application]. Moreover, there is no support for how exactly to correlate the claimed limitation "an instruction of the program is executed using a plurality of the configurations" with the disclosure of U.S. Pat. No. 7,210,129. For example, it is unclear as to whether the instruction itself inside the WHILE loop would be executed using a plurality of configurations, or whether the instruction itself would be executed using one configuration, but because the instruction is inside a WHILE loop which requires another configuration to interpret the WHILE conditions, the instruction itself would then be considered to be executed using a plurality of configurations in total. Examiner generally notes that it is unclear as to how the disclosure of the U.S. Pat. No. 7,210,129 would be applied to the environment of the instant application wherein the traditional instruction is replaced by a configuration in the known sense, referred to as a CIW (a term not referenced in U.S. Pat. No. 7,210,129).

9. Claim 23 recites the limitation "transmitting data directly from any one of the data processing cells to any other of the data processing cells" in lines 7-8. However, the original disclosure does not appear to support this concept of transmitting data "directly" from "any one" of the data processing cells to "any other" of the data processing cells.

10. Claim 24 recites the limitation "for each of at least one of the configurations, a plurality of instructions of the program are executable via a single instance of the respective configuration" in lines 14-16. However, the original disclosure does not appear to disclose this limitation (e.g. page 5, lines 30-32, discloses that the traditional instruction [and not a plurality of instructions] is replaced by a configuration in the known

sense, referred to in the following discussion as a complex instruction word). Examiner recommends that applicant specifically point out where and how in the original disclosure this limitation is supported in the event that Applicant disagrees with the rejection.

In response to examiner's previous recommendation that applicant specifically point out where and how in the original disclosure this limitation is supported in the event that applicant disagrees with the rejection, applicant argues that:

"The present application, at par. 37 of the application publication no. 2006/0090062, incorporates by reference internal docket number PACT11, which includes U.S. Pat. App. Ser. No. 09/967,847 (issued as U.S. Pat. No. 7,210,129), which claims priority to German Pat. App. No. DE 101 39 170, which is itself separately referred to and incorporated by reference at par. 3 of the application publication. (See also U.S. Pat. App. Pub. No. 2009/0006895, which explicitly states that PACT11 includes U.S. Pat. App. Ser. No. 09/967,847.) Support for the above-noted features of claims 23 and 24, respectively, may be found, for example, in U.S. Pat. No. 7,210,129 at 3:16-19, 3:33-35, 4:23-25, 6:41-42, 7:51- 58, and 11:1-6."

However, there are numerous deficiencies in this argument. First, the original disclosure does not appear to provide support for applicant's argument that PACT 11 "includes U.S. Pat. App. Ser. No. 09/967,847 (issued as U.S. Pat. No. 7,210,129)" because the IDS dated 7/16/2004 merely correlates "PACT11" to "DE 101 39 170.6". Applicant cites U.S. Pat. App. Pub. No. 2009/0006895 as explicitly stating that PACT11 includes U.S. Pat. App. Ser. No. 09/967,847. However, this explicit statement is not

part of the original disclosure. Examiner further generally notes the confusion that arises in assigning same "internal reference numbers" to different references in different pending patent applications.

Second, examiner notes that "DE 101 39 170.6" does not appear to provide support for the limitation in question. Examiner's conclusion is based on a translation of the entirety of "DE 101 39 170.6", included with the current office action. Given that more than 30 DE references are incorporated by reference, examiner respectfully asks applicant to provide translations and specific citations for any DE references relied upon to support subject matter in order to further compact prosecution. Examiner further generally notes that in any application which is to issue as a U.S. patent, essential material (defined as that which is necessary to (1) describe the claimed invention, (2) provide an enabling disclosure of the claimed invention, or (3) describe the best mode) may not be incorporated by reference to (1) patents or applications published by foreign countries or a regional patent office, (2) non-patent publications, (3) a U.S. patent or application which itself incorporates "essential material" by reference, or (4) a foreign application.

Finally, examiner notes that U.S. Pat. No. 7,210,129 does not appear to provide support for the limitations (even though U.S. Pat. No. 7,210,129 is not properly incorporated by reference, as detailed above). For example, while U.S. Pat. No. 7,210,129 may disclose "a configuration represents a plurality of instructions" in col. 3, lines 32-33, this disclosure is in the context of the invention of U.S. Pat. No. 7,210,129, and there is no linking support that an instruction of the program [of the instant

application] is executed using a plurality of the configurations [of the instant application]. Examiner generally notes that it is unclear as to how the disclosure of the U.S. Pat. No. 7,210,129 would be applied to the environment of the instant application wherein the traditional instruction is replaced by a configuration in the known sense, referred to as a CIW (a term not referenced in U.S. Pat. No. 7,210,129).

11. Claim 24 recites the limitation "transmitting data directly from any one of the data processing cells to any other of the data processing cells" in lines 7-8. However, the original disclosure does not appear to support this concept of transmitting data "directly" from "any one" of the data processing cells to "any other" of the data processing cells.

12. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

13. Claims 7-13, 15-17, and 19-22 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

14. Claim 7 recites the limitation "wherein at least one of: (a) a single function of the program is executed via sequential execution of a plurality of the set of configurations; and (b) for each of at least one of the set of configurations, a plurality of functions of the program are executed in sequence via execution of a single instance of the respective configuration" in lines 29-34. However, it is indefinite as to what is being conveyed. For example, the aforementioned limitation has the structure "wherein at least one of

Scenario A or Scenario B". This structure only conveys a fragment of a concept (e.g. is at least one of Scenario A or Scenario B occurring or not occurring?). Along those same lines, it is indefinite if at least one of Action A or Action B occurs, or if, in the case of Action A, at least one of a single function of the program [out of multiple functions of the program] is executed via sequential execution of a plurality of a set of configurations.

b. Claims 8-13, 15-17, and 19-22 are rejected for failing to alleviate the rejection of claim 7 above.

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 7, 10-11, 15-17, and 19-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (Smith) (US 6658564 B1) in view of Dockser (US 5860119) in view of Borkenhagen et al. (Borkenhagen) (US 6076157) in view of Bondalapati et al. (Bondalapati) (Reconfigurable Meshes: Theory and Practice).

17. Consider claim 7, Smith discloses a reconfigurable field of data processing cells (col. 3, lines 2-3, array of programmable logic regions) that are interconnected by a reconfigurable structure (col. 3, line 56, a routing structure) that are sharable over a plurality of configurations and that are configured for transmitting data directly from any

one o the data processing cells to any other of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; col. 6, line 66 to col. 7, line 1, discloses of run-time swapping of programmable logic device configuration data); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); determining, for a reconfigurable field of data processing cells (col. 8, lines 52-53, programmable logic resources), a set of configurations of the reconfigurable field of data processing cells, with respect to at least one of a function and an interconnection of the reconfigurable field of data processing cells, with execution of which configurations the program is run (col. 11, lines 60-63, compiling hardware functions into configuration patterns using a hardware description language compiler); a software compiler (col. 11, lines 62-63, hardware description language compiler 84); wherein the reconfigurable field of data processing cells is reconfigured with a different configuration (col. 6, lines 53-56, the virtual logic manager must therefore manage the run-time swapping of functions to be implemented in programmable logic); executing the configurations (col. 10, lines 50-51, executing on a reconfigurable hardware architecture), the executing including (a) configuring functions of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the

configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions), (b) configuring an interconnection of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; the routing structure is the interconnection); and during the executing: storing, in the data stream memory, at least one of the data stream and parts of the data stream (col. 4, lines 22-33, disclose of the random-access memory devices, it is inherent that they may be written to, the data with which it is written constitutes all or part of a data stream), wherein the data stream memory stores at least one vector (it is inherent that a data stream memory holds vectors of bits, such as each addressable line); wherein at least one of: (a) a single function of the program is executed via sequential execution of a plurality of the set of configurations; and (b) for each of at least one of the set of configurations, a plurality of functions of the program are executed in sequence via execution of a single instance of the respective configuration (col. 13, lines 28-29, plurality of blocks of configuration data that make up a given function).

However, Smith does not explicitly disclose that the data stream memory is a register which is operated as a FIFO memory. Smith also does not disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and in response to lapse of which the reconfigurable field of data

processing cells is reconfigured with a different configuration; and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith also does not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Dockser does disclose a register which is operated as a FIFO memory to process a data stream (e.g. col. 12, line 16-17, FIFO registers).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that one of many motivations of having a register act as a memory would be to allow quick access to data, and a FIFO buffer preserves order of incoming data (Dockser, col. 1, lines 63-65). In addition, Dockser makes a FIFO system both simple and inexpensive to implement (Dockser, col. 4, lines 6-40), despite decreases in management overhead.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dockser with the invention of Smith in order to allow quicker access to the data stream while preserving order of incoming data in a simple and inexpensive manner.

However, neither Smith nor Dockser disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and in response to lapse of which the reconfigurable field of data processing cells is

reconfigured with a different configuration, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith and Dockser also do not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Borkenhagen does disclose of determining, by software (col. 15, lines 20-22, the thread switch time-out register can be written by the processor itself with software code) and for each thread (col. 14, lines 65-67, each thread need not have the same thread switch time-out value), a respective maximum allowed execution runtime (col. 15, line 1, thread switch time-out value) prior to lapse of which a respective thread is uninterruptible (col. 14, line 45-48 discloses of the use of the time-out register so that an external interrupt is serviced within a limited period of time; thus the interrupt is not serviced until the time-out register indicates a time-out and not before), and in response to lapse of which a different thread is executed (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced) and for each thread, monitoring the respective maximum allowed execution runtime in order to interrupt the thread if the respective maximum allowed execution runtime is exceeded (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced).

Borkenhagen's teaching of interrupting after the maximum allowed execution runtime prevents processor hangs (Borkenhagen, col. 5, lines 35-37). It would also be

readily recognized that the use of different thread switch time-out values for different threads increases flexibility, and interrupting only after the time-out enables forward progress.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith and Dockser in order to prevent processor hangs in a flexible manner while enabling forward progress. Note that the overall combination results in Borkenhagen's specific thread switching policy being applied to Smith's configurations (and Smith's generic configuration switching policy) to result in an overall specific configuration switching policy, and Borkenhagen's teaching of using software to determine a thread-switch time-out value being applied to Smith's invention which uses a software compiler to result in the determination by a software compiler of a respective maximum allowed execution runtime.

However, Smith, Dockser, and Borkenhagen do not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Bondalapati discloses that the reconfigurable field of data processing cells (e.g. page 2, FPGA devices) are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources (page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation

because of the two dimensional topology, low-pin requirement and highly regular structure).

Bondalapati's teaching is very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure (Bondalapati, page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bondalapati with the invention of Smith, Dockser, and Borkenhagen in order to be very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure.

18. Consider claim 10, Smith discloses that the register is a RAM PAE (col. 4, lines 22-33, disclose of the random-access memory devices).

19. Consider claim 11, the combination of Smith, Dockser, and Borkenhagen discloses using the register to provide read and write access (Dockser, col. 4, lines 32-35, receive mode and transmit mode, and col. 5, lines 56-65, read and write pointers) when a virtual FIFO dividing line is implemented (Dockser, col. 3, lines 10-30, lines 54-56; the last word flag and end-of-packet detection means correlate to the said virtual

FIFO dividing line), wherein the program includes a multitask application, and a register is used for execution of at least one of two different tasks of the multitask application (Smith and Borkenhagen discloses of multitasking as cited in the independent claim; alternatively, multiple tasks within a thread).

20. Consider claim 15, Borkenhagen discloses a watchdog is used to recognize an exceedance of each respective maximum allowed execution runtime (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch).

21. Consider claim 16, Borkenhagen discloses that any one of the configurations that exceeds its respective maximum allowed execution runtime is treated as illegal (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; thus it is illegal for the first thread to continue executing).

22. Consider claim 17, Borkenhagen discloses that any one of the configurations that exceeds its respective maximum allowed execution runtime is treated as illegal (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; thus it is illegal for the first thread to continue executing).

23. Consider claim 19, Smith and Borkenhagen discloses an operating system performs a predefined step in response to an exceedance by a configuration of the

configuration's maximum allowed execution (Borkenhagen, col. 15, lines 1-19, thread switch time-out values forces a thread switch; col. 17, lines 41-42, operating system; Smith discloses in col. 8, lines 26-27, 52-53, discloses of operating systems allocating programmable logic resources to functions).

24. Consider claim 20, Smith discloses at least one of the configurations calls another of the configurations as a sub-routine (col. 12, lines 1-5 for example, a main function calls a dynamically-linked function).

25. Consider claim 21, Borkenhagen discloses the watchdog signal initiates a system trap (col. 15, lines 1-3, thread switch time-out values from the thread switch time-out register forces a thread switch; in other words, the thread is interrupted by the thread switching system).

26. Consider claim 22, Smith and Borkenhagen discloses, in response to the system trap, an operating system performs steps defined for a response to an invalid instruction (Borkenhagen, col. 15, lines 1-19, thread switch time-out values forces a thread switch; col. 17, lines 41-42, operating system; Smith discloses in col. 8, lines 26-27, 52-53, discloses of operating systems allocating programmable logic resources to functions).

27. Consider claim 23, Smith discloses a reconfigurable field of data processing cells (col. 3, lines 2-3, array of programmable logic regions) that are interconnected by a

reconfigurable structure (col. 3, line 56, a routing structure) that are sharable over a plurality of configurations and that are configured for transmitting data directly from any one o the data processing cells to any other of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; col. 6, line 66 to col. 7, line 1, discloses of run-time swapping of programmable logic device configuration data); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); determining, for the reconfigurable field of data processing cells (col. 8, lines 52-53, programmable logic resources), a set of configurations of the reconfigurable field of data processing cells, with respect to at least one of a function and an interconnection of the reconfigurable field of data processing cells (col. 11, lines 60-63, compiling hardware functions into configuration patterns using a hardware description language compiler), wherein an instruction of the program is executed using a plurality of the configurations (col. 13, lines 28-29, plurality of blocks of configuration data that make up a given function), the reconfigurable field being reconfigured between the use of different ones of the plurality of configurations (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store

and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; col. 6, line 66 to col. 7, line 1, discloses of run-time swapping of programmable logic device configuration data; see, for example, col. 10, lines 43-48, which discloses the virtual logic manager may swap blocks of configuration data in a particular programmable logic resource if such resources are limited. Furthermore, as shown in Fig. 5B, the VC-OS may allocate one programmable logic device resource to multiple blocks of configuration data); a software compiler (col. 11, lines 62-63, hardware description language compiler 84); implementing the configurations (col. 10, lines 50-51, executing on a reconfigurable hardware architecture); and during the implementing: configuring functions of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions), configuring an interconnection of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; the routing structure is the interconnection); storing, in the data stream memory, at least one of the data stream and parts of the data stream (col. 4, lines 22-33, disclose of the random-access memory devices, it is inherent that they may be written to, the data with which it is written constitutes all or part of a data

stream), wherein the data stream memory stores at least one vector (it is inherent that a data stream memory holds vectors of bits, such as each addressable line).

However, Smith does not explicitly disclose that the data stream memory is a register which is operated as a FIFO memory. Smith also does not disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith also does not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Dockser does disclose a register which is operated as a FIFO memory to process a data stream (e.g. col. 12, line 16-17, FIFO registers).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that one of many motivations of having a register act as a memory would be to allow quick access to data, and a FIFO buffer preserves order of incoming data (Dockser, col. 1, lines 63-65). In addition, Dockser makes a FIFO system both simple and inexpensive to implement (Dockser, col. 4, lines 6-40), despite decreases in management overhead.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dockser with the invention of Smith in

order to allow quicker access to the data stream while preserving order of incoming data in a simple and inexpensive manner.

However, neither Smith nor Dockser disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith and Dockser also do not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Borkenhagen does disclose of determining, by software (col. 15, lines 20-22, the thread switch time-out register can be written by the processor itself with software code) and for each thread (col. 14, lines 65-67, each thread need not have the same thread switch time-out value), a respective maximum allowed execution runtime (col. 15, line 1, thread switch time-out value) prior to lapse of which a respective thread is uninterruptible (col. 14, line 45-48 discloses of the use of the time-out register so that an external interrupt is serviced within a limited period of time; thus the interrupt is not serviced until the time-out register indicates a time-out and not before), and for each thread, monitoring the respective maximum allowed execution runtime in order to interrupt the thread if the respective maximum allowed execution runtime is exceeded (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced).

Borkenhagen's teaching of interrupting after the maximum allowed execution runtime prevents processor hangs (Borkenhagen, col. 5, lines 35-37). It would also be readily recognized that the use of different thread switch time-out values for different threads increases flexibility, and interrupting only after the time-out enables forward progress.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith and Dockser in order to prevent processor hangs in a flexible manner while enabling forward progress. Note that the overall combination results in Borkenhagen's specific thread switching policy being applied to Smith's configurations (and Smith's generic configuration switching policy) to result in an overall specific configuration switching policy, and Borkenhagen's teaching of using software to determine a thread-switch time-out value being applied to Smith's invention which uses a software compiler to result in the determination by a software compiler of a respective maximum allowed execution runtime.

However, Smith, Dockser, and Borkenhagen do not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Bondalapati discloses that the reconfigurable field of data processing cells (e.g. page 2, FPGA devices) are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources (page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and

a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Bondalapati's teaching is very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure (Bondalapati, page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bondalapati with the invention of Smith, Dockser, and Borkenhagen in order to be very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure.

28. Consider claim 24, Smith discloses a reconfigurable field of data processing cells (col. 3, lines 2-3, array of programmable logic regions) that are interconnected by a reconfigurable structure (col. 3, line 56, a routing structure) that are sharable over a plurality of configurations and that are configured for transmitting data directly from any one o the data processing cells to any other of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the

configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; col. 6, line 66 to col. 7, line 1, discloses of run-time swapping of programmable logic device configuration data); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); providing a program corresponding to a sequence of compilable high-level language instructions (col. 10, lines 49-50, high-level design specification or algorithm); determining, for the reconfigurable field of data processing cells (col. 8, lines 52-53, programmable logic resources), a set of configurations of the reconfigurable field of data processing cells, with respect to at least one of a function and an interconnection of the reconfigurable field of data processing cells, with execution of which configurations the program is run (col. 11, lines 60-63, compiling hardware functions into configuration patterns using a hardware description language compiler), wherein, for each of at least one of the configurations, a plurality of instructions of the program are executable via single instance of the respective configuration (col. 13, lines 33-34, a single block of configuration data that makes up a given function); a software compiler (col. 11, lines 62-63, hardware description language compiler 84); executing the configurations (col. 10, lines 50-51, executing on a reconfigurable hardware architecture); and during the executing: configuring functions of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input

and output regions, and the memory regions); configuring an interconnection of at least a subset of the data processing cells (col. 3, lines 60-65, the logical functions performed by a programmable logic device are determined by the configuration data stored in the configuration store and the connection of the configuration store to the logic regions, the routing structure, the input and output regions, and the memory regions; the routing structure is the interconnection); storing, in the data stream memory, at least one of the data stream and parts of the data stream (col. 4, lines 22-33, disclose of the random-access memory devices, it is inherent that they may be written to, the data with which it is written constitutes all or part of a data stream), wherein the data stream memory stores at least one vector (it is inherent that a data stream memory holds vectors of bits, such as each addressable line).

However, Smith does not explicitly disclose that the data stream memory is a register which is operated as a FIFO memory. Smith also does not disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith also does not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Dockser does disclose a register which is operated as a FIFO memory to process a data stream (e.g. col. 12, line 16-17, FIFO registers).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that one of many motivations of having a register act as a memory would be to allow quick access to data, and a FIFO buffer preserves order of incoming data (Dockser, col. 1, lines 63-65). In addition, Dockser makes a FIFO system both simple and inexpensive to implement (Dockser, col. 4, lines 6-40), despite decreases in management overhead.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dockser with the invention of Smith in order to allow quicker access to the data stream while preserving order of incoming data in a simple and inexpensive manner.

However, neither Smith nor Dockser disclose determining, by a software compiler and for each of the configurations, a respective maximum allowed execution runtime prior to lapse of which the respective configuration is uninterruptible, and for each configuration, monitoring the respective maximum allowed execution runtime in order to interrupt processing with the configuration if the respective maximum allowed execution runtime is exceeded. Smith also does not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Borkenhagen does disclose of determining, by software (col. 15, lines 20-22, the thread switch time-out register can be written by the processor itself

with software code) and for each thread (col. 14, lines 65-67, each thread need not have the same thread switch time-out value), a respective maximum allowed execution runtime (col. 15, line 1, thread switch time-out value) prior to lapse of which a respective thread is uninterruptible (col. 14, line 45-48 discloses of the use of the time-out register so that an external interrupt is serviced within a limited period of time; thus the interrupt is not serviced until the time-out register indicates a time-out and not before), and for each thread, monitoring the respective maximum allowed execution runtime in order to interrupt the thread if the respective maximum allowed execution runtime is exceeded (col. 15, lines 1-7, thread switch time-out value is decremented, and when it is determined to equal zero, a thread switch is forced).

Borkenhagen's teaching of interrupting after the maximum allowed execution runtime prevents processor hangs (Borkenhagen, col. 5, lines 35-37). It would also be readily recognized that the use of different thread switch time-out values for different threads increases flexibility, and interrupting only after the time-out enables forward progress.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Borkenhagen with the invention of Smith and Dockser in order to prevent processor hangs in a flexible manner while enabling forward progress. Note that the overall combination results in Borkenhagen's specific thread switching policy being applied to Smith's configurations (and Smith's generic configuration switching policy) to result in an overall specific configuration switching policy, and Borkenhagen's teaching of using software to determine a thread-

switch time-out value being applied to Smith's invention which uses a software compiler to result in the determination by a software compiler of a respective maximum allowed execution runtime.

However, Smith, Dockser, and Borkenhagen do not disclose that the reconfigurable field of data processing cells are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources.

On the other hand, Bondalapati discloses that the reconfigurable field of data processing cells (e.g. page 2, FPGA devices) are interconnected by a reconfigurable multi-dimensional bus structure that includes dynamically allocatable bus resources (page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Bondalapati's teaching is very attractive in terms of implementation because of the two dimensional topology, low pin requirement, and highly regular structure (Bondalapati, page 9, section 4, lines 1-3, the reconfigurable mesh has nearly constant diameter and a dynamically reconfigurable bus system. It is very attractive in terms of implementation because of the two dimensional topology, low-pin requirement and highly regular structure).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bondalapati with the invention of Smith, Dockser, and Borkenhagen in order to be very attractive in terms of implementation

because of the two dimensional topology, low pin requirement, and highly regular structure.

29. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Dockser, Borkenhagen, and Bondalapati as applied to claim 7 above, and further in view of Panwar et al. (Panwar) (US 5941977).

30. Consider claim 8, Smith, Dockser, Borkenhagen, and Bondalapati do not disclose at least one: I) of a register allocation device to allocate the register, and ii) a register releasing device to release the register.

On the other hand, Panwar does disclose at least one: I) of a register allocation device to allocate the register, and ii) a register releasing device to release the register (col. 7, lines 31-39, register window allocation and col. 7, lines 54-64, register management).

Panwar's teaching enables processes to access registers independent of other processes executing within the processor (Panwar, col. 7, lines 35-38).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Panwar with the invention of Smith, Dockser, Borkenhagen, and Bondalapati in order to access registers independent of other processes executing within the processor.

31. Consider claim 9, Panwar discloses that the register allocation device is preserved over multiple reconfigurations of the reconfigurable field of data processing

cells (col. 2, lines 25-42, col. 6, lines 32-36, col. 7, lines 31-39 and 54-64; the multithreading aspect in which each thread has its corresponding registers conserved correlates to the different reconfigurations as per Smith's correlation between configurations and threads as explained above).

32. Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Dockser, Borkenhagen, and Bondalapati as applied to claim 7 above, and further in view of Davis et al. (Davis) (US PAT 4041462).

33. Consider claim 12, Smith, Dockser, Borkenhagen, and Bondalapati do not explicitly disclose at least one memory unit configured for use as a stack and being configured to indicate at least one of a stack underflow state and a stack overflow state.

On the other hand, Davis does disclose at least one memory unit configured for use as a stack and being configured to indicate at least one of a stack underflow state and a stack overflow state (col. 14, lines 1-4, limit checking facilities which test for overflow and underflow, and lines 21-32, PSW)

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that stacks in general are an easily implemented method of dynamic allocation of storage space for data, and a simple efficient mechanism for enqueuing data and/or parameters.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the stacks of Davis with the invention of Smith, Dockser, Borkenhagen, and Bondalapati in order to easily implement a method of

dynamic allocation of storage space for data, and efficiently enqueue data and/or parameters.

34. Consider claim 13, the claim is rejected for the same reasons as claim 12 above. In addition, Davis discloses the at least one of the underflow state and overflow state is of an operating system unit (col. 14, lines 1-4 and lines 21-32; also, note the PSW is typically accessed by the operating system).

Response to Arguments

35. Applicant on page 7 first appropriately characterizes the Borkenhagen reference as referring to a maximum runtime for a thread, and not for a configuration. Applicant then argues that Smith's teaching of a configuration and of a thread would not suggest applying Borkenhagen's teaching of a maximum runtime to a configuration; this argument is addressed below.

36. Applicant argues on page 7 that, thus, there is no one-to-one correspondence of a thread to a configuration, and the reference to a forced thread switch after some time in the Borkenhagen reference in no way suggests a forced configuration switch after some time.

However, Smith discloses that a function can be executed either in software or hardware (e.g. col. 7, lines 7-8). Functions executed in software are compiled into threads, and functions executed in hardware are compiled into configurations (e.g. col.

11, lines 59-63). Thus, there is a correspondence between the threads and configurations: each performs the function, albeit one in software and one in hardware. It would be readily recognized to one of ordinary skill in the art at the time of the invention that the motivation of Borkenhagen is applicable and beneficial regardless of whether functions are executed in software or hardware.

Examiner notes that Smith discloses that a typical scheduling system for a reconfigurable computer may use a time-multiplexing system in which programmable logic resources may be allocated to application functions and which may involve switching between different functions at predetermined time periods (col. 8, line 66 through col. 9, line 4). Therefore, applicant's contention that one of ordinary skill in the art would only apply multitasking teachings to threads and not to configurations is inaccurate at least because Smith already does so, though as examiner has pointed out, the correlation between the threads and configurations would be enough to motivate the combination. Nevertheless, it is Borkenhagen's multitasking policy in particular which examiner is applying to Smith. Again, given that Smith not only correlates threads to configurations but also teaches of applying the multitasking aspect of threads to configurations, one of ordinary skill in the art at the time of the invention would recognize that Borkenhagen's specific multitasking policy would not only be applicable to threads but also to configurations, given that the goal of the policy to prevent processor hangs is applicable regardless of whether the program which is hanging is being executed in software or hardware. The execution of functions in hardware (e.g. to provide faster execution times as per Smith, col. 1, lines 46-47) does

not nullify the benefit of Borkenhagen's teaching. It would be readily recognized that an explicit recitation that any general modification that one can do to threads, one can likewise do to configurations, is not necessary for the aforementioned combination to nevertheless be proper.

37. Applicant argues on page 8 that at column 8, line 66 to column 9, line 4, the Smith reference merely suggests applying a time-multiplexing system to functions, and that nowhere does the Smith reference suggest applying a time-multiplexing system to configurations.

However, Smith discloses in numerous places discloses of the correlation between functions and configurations. For example, col. 11, lines 60-63 discloses of compiling hardware functions into configuration patterns using a hardware description language compiler, and col. 9, lines 7-11 discloses of function prefetching wherein the programmable logic resource can be loaded with the configuration data before the function is required. In other words, the invention of Smith entails functions which correlate to configuration data. As another example, col. 13, lines 31-33, discloses of a single block of configuration data making up a given function. Therefore, any extent by which Smith suggests applying a time-multiplexing system to functions would likewise apply to configuration data, as functions which are time-multiplexed and implemented via configurations consequently entails the configurations being time-multiplexed.

38. Applicant argues on page 8 that while the Smith reference may provide certain configurations of hardware that provide for operation in a manner that corresponds in its entirety to a function as a whole, the time-multiplexing is ultimately provided on a function-by-function basis, and not a configuration-by-configuration basis. Applicant provides as an example, if a configuration is usable for multiple functions, then, while the time-multiplexing may provide for interrupting a particular function, the configuration may continue to be used without reconfiguration.

However, examiner notes that the instant disclosure does not disclose of a configuration usable for multiple functions or its corresponding operation, and thus it is not necessarily the case in the hypothetical scenario that the configuration may continue to be used without reconfiguration: it could also be the case that reconfiguration occurs (e.g. because the system does not know configurations used for multiple functions are the same, or because reloading is necessary to reinitialize the configuration for use by another function, or so forth). Moreover, it is not necessarily the case that a configuration usable for multiple functions (i.e. two different functions which use the same configuration) would even be covered by Smith. In contrast, Smith discloses of time-multiplexing functions implemented via corresponding configurations (and thus discloses of time-multiplexing configurations), which supports the modification by Borkenhagen of a specific multitasking policy to configurations in particular.

In addition, examiner generally notes that the instant disclosure does not appear to explicitly disclose of the behavior that applicant argues contrasts with Smith, as page 7, lines 6-7 only discloses that the maximum execution time of a CIW has an upper limit.

There is no disclosure of "different functions" using a CIW such that a CIW is executed consecutively to carry out the different functions, for example.

39. Applicant argues on page 8 that the Smith reference still does not suggest applying time-multiplexing to a configuration.

However, Smith discloses in numerous places discloses of the correlation between functions and configurations. For example, col. 11, lines 60-63 discloses of compiling hardware functions into configuration patterns using a hardware description language compiler, and col. 9, lines 7-11 discloses of function prefetching wherein the programmable logic resource can be loaded with the configuration data before the function is required. In other words, the invention of Smith entails functions which correlate to configuration data. As another example, col. 13, lines 31-33, discloses of a single block of configuration data making up a given function. Therefore, any extent by which Smith suggests applying a time-multiplexing system to functions would likewise apply to configuration data, as functions which are time-multiplexed and implemented via configurations consequently entails the configurations being time-multiplexed.

40. Applicant argues on page 8 that claim has been amended to facilitate matters.

However, the amendment does not appear to have support in the original description. For a detailed analysis, see the 112 section above.

41. Applicant argues on page 9 that 10:14 et seq discloses that the plurality of blocks does not disclose a plurality of configurations, where reconfiguration is performed between the use of different configurations.

However, see, for example, col. 10, lines 43-48, which discloses the virtual logic manager may swap blocks of configuration data in a particular programmable logic resource if such resources are limited. Furthermore, as shown in Fig. 5B, the VC-OS may allocate one programmable logic device resource to multiple blocks of configuration data. In other words, the programmable logic device resource is reconfigured between uses of different blocks of configuration data.

Additionally, examiner generally notes that Smith's teaching of time-multiplexing further teaches the scenario of the reconfigurable field being reconfigured between the use of different ones of the plurality of configurations. Examiner also notes that the claimed limitation of "an instruction of the program is executed using a plurality of the configurations" remains subject to a written description issue.

42. Applicant argues on page 9 that col. 13, lines 33-34 of the Smith reference merely state that a single programmable logic resource may be allocated to a single block of configuration data that makes up a given function, but do not suggest that multiple instructions are executed using a single instance of a configuration.

However, the idea of a function being comprised of multiple instructions would have been obvious to one of ordinary skill in the art at the time of the invention. See for example, col. 11, lines 59-63, which discloses that software functions are compiled into

threads and hardware functions are compiled into configuration patterns. It would have been obvious to one of ordinary skill in the art at the time of the invention for a thread to be comprised of multiple instructions" (especially those compiled from "complex instructions" of col. 8, line 59). Therefore, a function which is made of multiple instructions, when executed via hardware such that a single programmable logic resource may be allocated to a single block of configuration data that makes up a given function, teaches the overall concept that multiple instructions are executed using a single instance of a configuration. Examiner also notes that the claimed limitation of "a plurality of instructions of the program are executable via a single instance of the respective configuration" remains subject to a written description issue.

43. The arguments with regard to the remaining claims rely on the arguments made with regard to claim 7; consequently, examiner's response to arguments with regard to claim 7 are likewise applicable to these remaining claims.

Conclusion

44. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

c. Kamiya et al. (US 20100306602) discloses of using a watchdog timer to detect abnormalities of a processing task of software executed by a CPU.

45. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KEITH VICARY whose telephone number is (571)270-1314. The examiner can normally be reached on Monday - Thursday, 7:00 a.m. - 5:30 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Keith Vicary/
Examiner, Art Unit 2183

/EDDIE CHAN/
Supervisory Patent Examiner, Art Unit 2183